

# po-4. 条件分岐, ステップ実行

トピックス: 条件分岐, if, else, ステップ実行  
(Python Tutor による演習)

URL: <https://www.kkaneko.jp/pro/po/index.html>

(Python プログラミングの基本)

金子邦彦





```
1 age = 10
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```



Print output (drag lower right corner to resize)

500

Frames

Objects

Global frame

age 10

条件分岐, `age <= 12` のときは 500,  
`age > 12` のときは 1200

Python 3.6

```
→ 1 age = 30
→ 2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
→ 5     print(1200)
```

ステップ実行により, ジャンプの様子を観察

# 全体まとめ



- **条件分岐**では、変数や式の値によって結果が変わるなどの判断を行う  
条件式は「**age <= 12**」のようになる
- プログラムは、通常は、上から順に1つずつ実行される（**逐次実行**）
- **条件分岐**や**繰り返し（ループ）**では、逐次実行とは違う実行の流れになる
- **ステップ実行**により、変数等の変化、プログラム実行の流れを観察できる

	項目
	復習
4-1	条件分岐
4-2	演習
4-3	条件分岐のステップ実行

# メソッドアクセス, 代入



## Python プログラムの例

```
x = 100
a = x + 200
enemy1 = hero.findNearestEnemy()
hero.attack(enemy1)
```

- **代入** : **オブジェクト名** + 「**=**」  
+ 式または値またはメソッド呼び出し
- **メソッドアクセス** : **オブジェクト名** + 「**.**」  
+ **メソッド名** + 「**()**」 (引数を付けることも)

Python プログラムでは, その他にも, 属性アクセス, 関数呼び出し, 制御, 「\*」, 「+」などの演算子, コマンド, 定義など

# Python Tutor の起動



① **ウェブブラウザ**を起動する

② **Python Tutor** を使いたいので, 次の URL を開く  
**<http://www.pythontutor.com/>**

③ 「**Python**」 をクリック ⇒ **編集画面**が開く

## Learn Python, JavaScript, C, C++, and Java

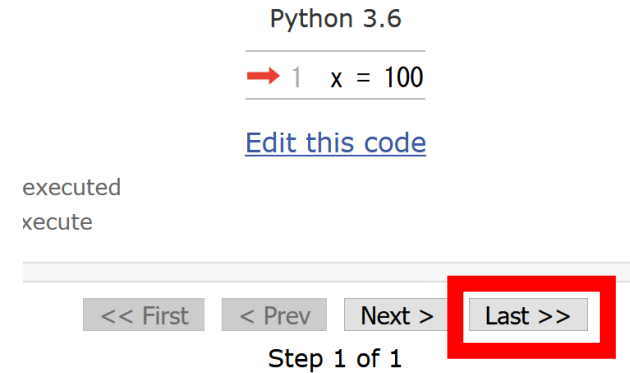
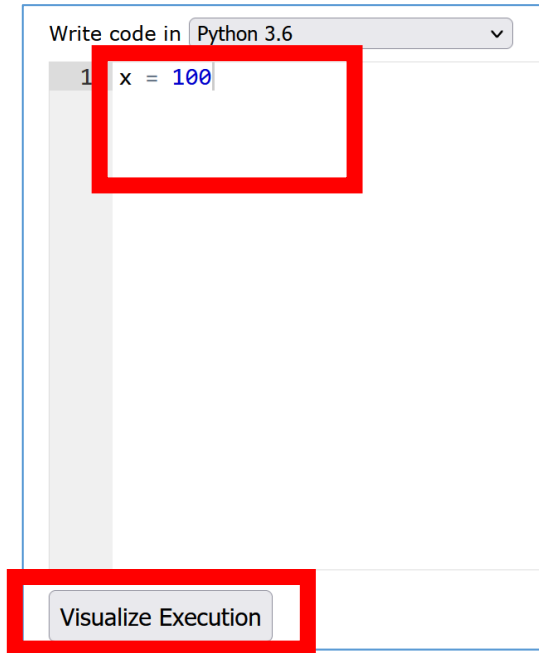
This tool helps you learn Python, JavaScript, C, C++, and Java programming by [visualizing code execution](#). You can use it to debug your homework assignments and as a supplement to online coding tutorials.

Start coding now in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

**Over 15 million people in more than 180 countries** have used Python Tutor to visualize over 200 million pieces of code. It is the most widely-used program visualization tool for computing education.

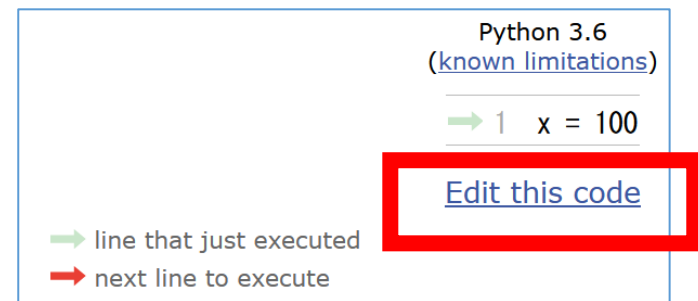
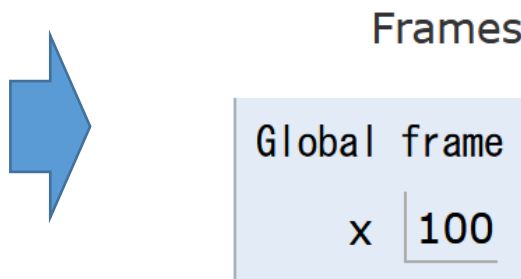
You can also embed these visualizations into any webpage. Here's an example showing recursion in Python:

# Python Tutor でのプログラム実行手順



(1) 「**Visualize Execution**」をクリックして**実行画面**に切り替える

(2) 「**Last**」をクリック。



(3) 実行結果を確認する。

(4) 「**Edit this code**」をクリックして**編集画面**に戻る

# Python Tutor 使用上の注意点①



- 実行画面で、次のような赤の表示が出ることもある →  
無視してよい

過去の文法ミスに関する確認表示  
邪魔なときは「Close」

## Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6  
([known limitations](#))

```
→ 1 x = 100
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 1 of 1

[Customize visualization](#)

Frames Objects

You just fixed the following error:

```
1 x = 100!
```

**SyntaxError: invalid syntax (<string>, line 1)**

Please help us improve this tool with your feedback.  
What misunderstanding do you think caused this error?

Submit Close [Hide all of these pop-ups](#)



# Python Tutor 使用上の注意点②



「please wait ... executing」のとき，10秒ほど待つ。



→ 混雑しているときは，「Server Busy・・・」  
というメッセージが出ることがある。

混雑している。少し（数秒から数十秒）待つと自動で表示が変わる（変わらない場合には，操作をもう一度行ってみる）

## 4-1. 条件分岐

# 条件分岐



変数の値によって, 結果が変わる (判断)

age の値が **12以下** → **500**  
**13以上** → **1800**

条件式は「**age <= 12**」のようになる

# 条件分岐のプログラム例



```
age = 18
```

```
if age <= 12:
```

```
    print(500)
```

```
else:
```

```
    print(1800)
```

if (age <= 12)の直後に「:」  
else の直後に「:」  
(どちらも, コロン)

**字下げも正確に！**

**print の前に, 「タブ (Tab)」を1つだけ**

## 4-2. 演習

# 演習

資料 : 15 ~ 18

## 【トピックス】

- 条件分岐
- if
- else

## ① Python Tutor のエディタで次のプログラムを入れる

```
1 age = 10
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

if (age <= 12)の直後に「:」  
else の直後に「:」  
(どちらも、コロン)

字下げも正確に！

print の前に、「タブ (Tab)」を1つだけ

```
1 age = 10
2 if age <= 12:
3     print(500)
4     else:
5         print(1200)
```

正しくない字下げ

「delキー」などを使い  
ながら編集

```
1 age = 10
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

正しい字下げ



② 実行するために、「**Visual Execution**」をクリック。そして「**Last**」をクリック。結果を確認

Print output (drag lower right corner to resize)

```
500
```

Frames

Objects

```
Global frame
```

```
age 10
```

結果の  
「500」を確認

- ③ 10 を **30** に書き換えて、実行し、結果を確認  
結果は **1800** である

結果の  
「**1200**」を確認

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

Print output (drag low

1200

Frames

Global frame

age 30

「**Visual Execution**」をクリック。そして「**Last**」をクリック。結果を確認。  
「**Edit this code**」をクリックすると、エディタの画面に戻る

## 4-3. 条件分岐のステップ実行

# プログラム実行

- **命令（コマンド）** は、順番に並んでいる
- 通常は、上から順に1つずつ実行される（逐次実行）
- **条件分岐**や**繰り返し（ループ）** では、逐次実行とは違う実行の流れになる

※ プログラミング言語の種類によっては、順番に並んでいないという場合もある： SQL, Scheme, Prolog など

# ステップ実行



- ステップ実行により、プログラム実行の流れをビジュアルに観察

Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

Print output (drag lower right)

Frames

Global frame

age 30

Step 3 of 3

<< First < Prev Next > Last >>

## 演習

資料：23 ~ 28

### 【トピックス】

- Python Tutor でのステップ実行の操作
- 変数の値の変化
- 実行の流れの変化（ジャンプ）

# ① まずは、「Visualize Execution」をクリックして、実行開始

Write code in Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
6
```

Help improve this tool by completing a [short user survey](#)

Visualize Execution

Live Programming Mode



すでに、プログラムを実行中だったときは、「**First**」をクリックして、最初の一行に戻すことができる

Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
→ 5     print(1200)
```

[Edit this code](#)

ited  
e

Navigation buttons: << First, < Prev, Next >, Last >>

Done running (3 steps)



- ② 「**Step 1 of 3**」と表示されているので、  
全部で、**ステップ数**は**3**あることが分かる  
(ステップ数と、プログラムの行数は**違うもの**)

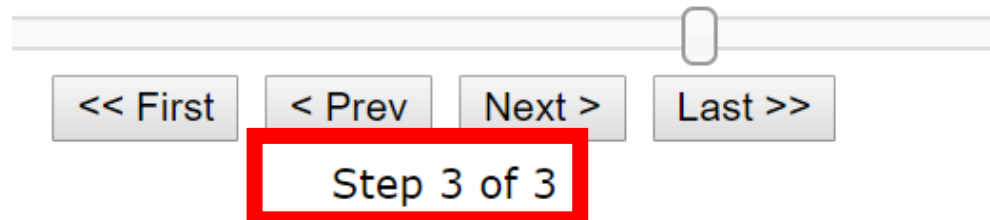
Python 3.6

```
1 age = 30  
→ 2 if age <= 12:  
3     print(500)  
4 else:  
→ 5     print(1200)
```

[Edit this code](#)

uted

ie



A progress bar with a slider and four navigation buttons: '<< First', '< Prev', 'Next >', and 'Last >>'. Below the buttons, the text 'Step 3 of 3' is displayed and highlighted with a red box.

③ **ステップ実行**したいので、「Next」をクリックしながら、矢印の動きを確認。

※「Next」ボタンを何度か押し、それ以上進めなくなったら終了

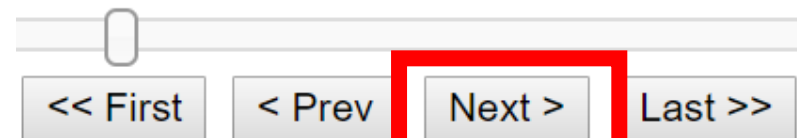
見どころ  
2行目から 5行目へ  
**ジャンプ**するところ



```
Python 3.6  
-----  
→ 1 age = 30  
→ 2 if age <= 12:  
   3     print(500)  
   4 else:  
   5     print(1200)  
-----
```

[Edit this code](#)

ed



Step 2 of 3



④ 最後の行まで達したら、「**Edit this code**」をクリックして、エディタの画面に戻る

Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

ited  
e

<< First

< Prev

Next >

Last >>

Done running (3 steps)

⑤ 次のように「**age = 10**」と書き換えて、いままでと同じことを行う

見どころ  
**3行目で実行が止まる**

Python 3.6

```
1 age = 10
2 if age <= 12:
→ 3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

uted  
te

<< First < Prev **Next >** Last >>

Done running (3 steps)

# 全体まとめ



- **条件分岐**では、変数や式の値によって結果が変わるなどの判断を行う  
条件式は「**age <= 12**」のようになる
- プログラムは、通常は、上から順に1つずつ実行される（**逐次実行**）
- **条件分岐**や**繰り返し（ループ）**では、逐次実行とは違う実行の流れになる
- **ステップ実行**により、変数等の変化、プログラム実行の流れを観察できる

# 演習問題

# 条件分岐



- 次のプログラムを作成

① **weight** と料金の関係は次の通り

**weight** の値が **100以下** → **0**

**100より大きい** → **1000**

② **weight = 80** に設定してテスト実行

# 正解の例



Python 3.6  
([known limitations](#))

---

```
1 weight = 80
2 if weight <= 100:
→ 3     print(0)
4 else:
5     print(1000)
```

---

[Edit this code](#)

Print output (drag lo

0

Frames

Global frame

weight | 80



# 全体まとめ



- **条件分岐**では、変数や式の値によって結果が変わるなどの判断を行う  
条件式は「**age <= 12**」のようになる
- プログラムは、通常は、上から順に1つずつ実行される（**逐次実行**）
- **条件分岐**や**繰り返し（ループ）**では、逐次実行とは違う実行の流れになる
- **ステップ実行**により、変数等の変化、プログラム実行の流れを観察できる

# Python 関連ページ



- Python まとめページ

<https://www.kkaneko.jp/tools/man/python.html>

- Python 入門（スライド資料とプログラム例）

<https://www.kkaneko.jp/pro/pf/index.html>

- Python プログラミングの基本（スライド資料とプログラム例）

<https://www.kkaneko.jp/pro/po/index.html>

- Python プログラム例

<https://www.kkaneko.jp/pro/python/index.html>

- 人工知能の実行（Google Colaboratory を使用）

<https://www.kkaneko.jp/ai/ni/index.html>

- 人工知能の実行（Python を使用）（Windows 上）

<https://www.kkaneko.jp/ai/deepim/index.html>